From Meaningful Orderings in the Web of Data to Multi-Level Pattern Structures

Quentin Brabant, Miguel Couceiro,
Amedeo Napoli, Justine Reynaud
LORIA (CNRS, Inria Nancy Grand Est, Université de Lorraine)
e-mails: name.surname@loria.fr

Abstract

We define a pattern structure whose objects are elements of a supporting ontology. In this framework, descriptions constitute trees, made of triples subject-predicate-object, and for which we provide a meaningful similarity operator. The specificity of the descriptions depends on a hyperparameter corresponding to their depth. This formalism is compatible with ontologies formulated in the language of RDF and RDFS and aims to set up a framework based on pattern structures for knowledge discovery in the web of data.

1 Introduction

The recent developpement of the web of data made available an increasing amount of ontological knowledge in various fields of interest. This knowledge is mainly expressed as set out in the specifications of the Resource Description Framework (RDF) [1] and RDF Schema (RDFS) [2].

This preliminary study aims to provide a knowledge discovery framework that produces an ordered structure summarizing and describing objects from a given ontology. The definition of ontology that we use can be seen as an abstraction of the data model defined by RDF and RDFS standards. In this paper, by an ontology we mean a structure that combines two types of knowledge about a set of objects. The first type is *subsumptional knowledge*: it corresponds to the ordering of objects in terms of their specificity, which is given by a subsumption relation (also sometimes referred to as *IS A* relation). The second type is *predicate knowledge*, thought of as a set of relations that occur between objects. The latter kind of knowledge can be naturally represented by an oriented labelled multigraph where nodes are objects and where the labels of arcs indicate the kind of relation that occurs between the origin and the target objects of the arc.

Our approach is rooted in Formal Concept Analysis (FCA) [7] and pattern structures [6], which are two frameworks for knowledge discovery that allow the comparison and the classification of objects with respect to their descriptions. In this article we propose a framework to integrate knowledge lying in ontologies into the knowledge discovery process. More precisely, we define a pattern

[1] https://www.w3.org/RDF/
[2] https://www.w3.org/TR/rdf-schema/

structure in which objects are described according to the information available in the ontology. This work follows recent propositions made in [1, 10], with the goal of extracting knowledge from the ontology with as much accuracy as possible.

In Section 2 we survey basic background on order theory and introduce the basic structures that constitute ontology in our framework. In Section 3 we recall basic notions and terminology of FCA and pattern structures. In Section 4 we define a pattern structure in which knowledge from the ontology is used to build the descriptions of objects. The potential of this framework in knowledge discovery is then briefly discussed in Section 5.

2 Preliminaries

2.1 Subsumption Relations and Posets

In this paper we will make use of several order relations on differents sets, that we think of as subsumption relations. In other words, they order objects from the most specific to the most general. With no danger of ambiguity, they all will be denoted by the same symbol $\sqsubseteq$. For a set X endowed with $\sqsubseteq$ and $a, b \in X$, the fact that a is subsumed by b will be denoted by $a \sqsubseteq b$, with the meaning that a is more specific than b or, equivalently, that b is more general than a.

The subsumption relations considered hereinafter have the property that any two elements from a given set have a unique *least common subsumer*, in other words, they constitute semilattice orders. This enables us to define a *similarity* (or *meet*) operator $\sqcap$ by, for $a, b \in X$, $a \sqcap b$ is the least common subsumer of a and b. Note that

$$a \sqsubseteq b \quad \Leftrightarrow \quad a \sqcap b = b. \tag{1}$$

Note that since we only consider finite sets, X necessarily has a most general element. As for subsumption relations, we will denote all similarity operators by the same symbol $\sqcap$ when there is no danger of ambiguity.

Also, for $A \subseteq X$, we will denote by $mins(A)$ the subset of minimal values of A, i.e.,

$$mins(A) = \{c \in A \mid \nexists c' \in A : c' \sqsubseteq c \text{ and } c \neq c'\}.$$

In this paper, we will use the term *partially ordered set* (or simply *poset*) for any set X of elements ordered by a subsumption relation $\sqsubseteq$.

2.2 Ontological Knowledge

In this section we aim to define an ontology that combines information of two kinds, namely *subsumptional knowledge* and *predicate knowledge* as explained below.

Throughout this paper we will consider two disjoint sets, a set V of *objects* and a set P of *predicates*. We will also consider a subsumption order on each of these sets, such that $(P, \sqsubseteq)$ and $(V, \sqsubseteq)$ are posets with most general elements denoted by $\top_P$ and $\top_V$, respectively.

Now let $\mathscr{G} = (V, E)$ be an oriented multigraph with labelled arcs, where V is a set of vertices and $E \subseteq V \times P \times V$ is a set of triples such that $(s, p, o) \in E$ if s is connected to o by an arc labelled by p. Note that an element $s \in V$ can

be connected to $o \in V$ by several arcs with different labels, however there can be at most one arc with a label p connecting s to o.

In our framework, $(P, \sqsubseteq)$ and $(V, \sqsubseteq)$ provide the *subsumptional knowledge* of the ontology while the graph $\mathscr{G}$ provides its *predicate knowledge*.

Example 1. The following example illustrates the kind of knowledge that can be captured by such structures. Abbreviated namings of objects and predicates are given between parentheses. Let

$$\begin{aligned} V =& \{\texttt{beef}, \texttt{dessert}, \texttt{dish}, \texttt{egg}, \texttt{ice-cream}(\texttt{i-c.}), \texttt{onion tortilla}(\texttt{o. tortilla}), \\ & \texttt{menu1}(\texttt{m}_1), \texttt{menu2}(\texttt{m}_2), \texttt{onion}, \texttt{potato}, \texttt{steak and chips}(\texttt{s\&c}), \\ & \texttt{strawberry i-c.}, \texttt{tortilla}, \texttt{vanillai-c.}, \top_{\texttt{V}}\}, \\ P =& \{\texttt{has component}(\texttt{hcomp}), \texttt{has ingredient}(\texttt{hing}), \top_{\texttt{P}}\}. \end{aligned}$$

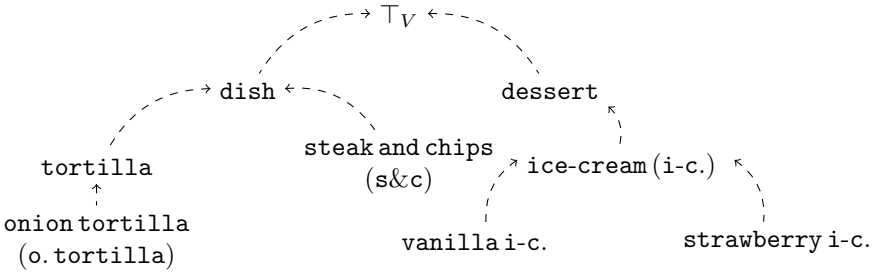

Figure 1: The poset of objects given by $(V, \sqsubseteq)$. Dashed arrows represent subsumption relations from their origin to their target. All elements of V that do not appear in the figure are subsumed by $\top_V$. This poset indicates for example that $\texttt{tortilla} \sqsubseteq \texttt{dish}$, and that $\texttt{tortilla} \sqcap \texttt{s\&c} = \texttt{dish}$.

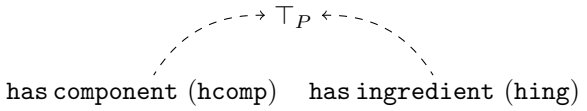

Figure 2: The poset of predicates given by $(P, \sqsubseteq)$. Dashed arrows represent a subsumption relation between their target and origin.

Figures 1, 2 and 3 represent respectively the posets of objects, the poset of predicates and the graph $\mathscr{G}$ describing relations between objects through predicates.

The pair of subsumptional and predicate knowledge is commonly used to describe objects within the same field of interest (e.g., class diagrams in object oriented programming). Ontologies using RDF and RDFS formats also describe these types of knowledge, but with a clear separation made between *classes* and *instances*. Indeed, RDF and RDFS specifications constitute a framework for describing *resources*, which are divided into three categories: classes, instances

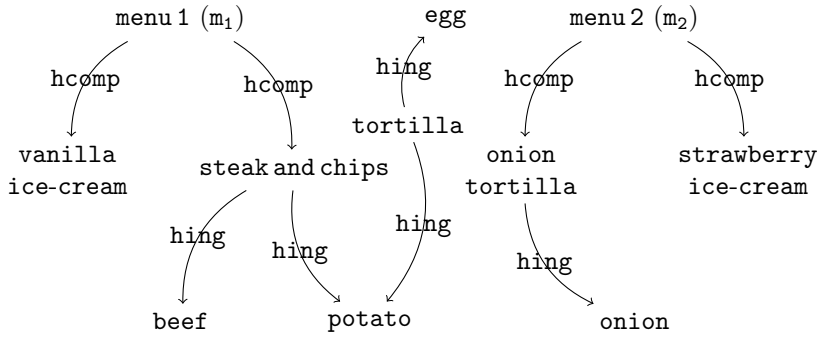

Figure 3: The relation graph $\mathscr{G}$. Vertices are elements of V, while arcs are labelled with predicates of P. Arcs provide information about the relations between vertices. This graph represents, for instance, the facts that tortilla is made of eggs and potatoes, and that menu 1 contains vanilla ice-cream. Isolated vertices (i.e. objects that share no arcs with any other object) are not represented in the graph.

and *properties*. Instances can be related to each other via properties, while classes and properties are organized into so-called *hierarchies*. Hierarchies, as well as relation between instances, are expressed through a set of triples of the form resource-property-resource. Class and properties hierarchies are built using the properties `rdfs:subClassOf` and `rdfs:subProperty`, respectively. Instances are affected to classes using the `rdf:type` property.

A natural correspondence between RDFS and our formalism is then the following. The graph $\mathscr{G}$ represents the set of RDF triples (except those triples describing hierarchical relations), the set V represents the set of all classes and instances in the ontology, and the subsumption relation on V summerizes the information given by `rdfs:subClassOf` and `rdf:type` properties. Moreover, $(P, \sqsubseteq)$ corresponds to the RDFS property hierarchy that is expressed through `rdfs:subPropertyOf`.

Remark 1. RDFS hierarchies are not necessarily semilattices, since two elements do not necessarily have a unique least common subsumer. However, it is always possible to embed a poset into a lattice of subsets (see, e.g., [3, 4]).

In the following section we briefly recall the basics of FCA and pattern structures, which can be used to explore ontological knowledge.

3 FCA and Pattern Structures

Formal Concept Analysis (FCA) is a mathematical framework used in classification and knowledge discovery. The basic framework considers a set of objects, a set of attributes, and an incidence relation between the two. More precisely a *formal context* is a triple (G, M, I) where G is the set of objects, M is the set of attributes and $I \subseteq G \times M$ is an incidence relation between G and M. Here $(g, m) \in I$ is interpreted as the object g has the attribute m. In this setting two

derivation operators $.' : 2^G \to 2^M$ and $.' : 2^M \to 2^G$ are usually defined by

$$A' = \{m \in M \mid \forall g \in A : (g, m) \in I\},$$
$$B' = \{g \in G \mid \forall m \in B : (g, m) \in I\},$$

for every $A \subseteq G$ and $B \subseteq M$. A *formal concept* is then a pair (A, B) such that $A = B'$ and $B = A'$. In other words, A is the set of all objects that possess all attributes in B and, dually, B is set of all attributes common to all objects in A. The concept lattice associated to the formal context is then the set of all formal concepts ordered by

$$(A_1, B_1) \leq (A_2, B_2) \quad \text{if and only if} \quad A_1 \subseteq A_2 \quad \text{or, equivalently,} \quad B_2 \subseteq B_1.$$

FCA is fully detailed in [7].

FCA only deals with binary attribute values, i.e. objects are described by binary $|M|$-sequences where each 0/1 component expresses the presence/absence of the corresponding attribute in the object. In the pattern structure framework, these descriptions become richer, the so-called *patterns*, and they may take values other than 0/1.

In this framework, the set of patterns is usually denoted by D and endowed with a similarity operator $\sqcap$ that defines a semilattice $\underline{D} = (D, \sqcap)$. Note that with this similarity operator can define a subsumption relation $\sqsubseteq$ given by (1). In this sense we say that a decription d is more *general* than a description d' if $d' \sqsubseteq d$. Descriptions of objects $g \in G$ are given by a mapping $\delta : G \to D$. The corresponding pattern structure is then $(G, \underline{D}, \delta)$.

As in FCA, two derivation operators are considered, namely, $.^\diamond : 2^G \to D$ and $.^\diamond : D \to 2^G$, and defined by

$$A^\diamond = \prod_{g \in A} \delta(g) \quad \text{and} \quad d^\diamond = \{g \in G \mid d \sqsubseteq \delta(g)\},$$

for every $A \subseteq G$ and every $d \in D$. A *pattern concept* is then a pair (A, d) such that $A = d^\diamond$ and $d = A^\diamond$. The set A is the set of objects whose descriptions are more general than d, and d is the similarity of descriptions of objects in A.

Note that a pattern structure can be translated into the FCA formalism [6]. A representation context of a pattern structure $(G, \underline{D}, \delta)$ is a formal context (G, M, I) where the set of attributes M is a subset of the set of patterns D, and the incidence relation I is defined by: $(g, m) \in I$ if $m \sqsubseteq \delta(g)$. A representation context can be seen as a formal context where patterns are translated in terms of binary attributes values.

4 A Pattern Structure for Objects in an Ontology

In this section we discuss the construction of a pattern structure whose objects are the vertices of $\mathscr{G}$, that is, where $G \subseteq V$ (see Section 2).

4.1 Simple Descriptions

The description of an object $g \in G$ is an element of the set of patterns that represents the characteristics of g. To give a meaningful description to g, we

make use of the two types of knowledge present in the ontology, described in Section 2, namely, subsumptional knowledge which is given by the position of g in the poset, and the predicate knowledge which is provided by the multigraph $\mathscr{G}$.

Example 2. Consider the two structures given in Example 1. From the poset of objects we know that "onion tortilla" is a tortilla, a dish, and a thing ($\top_V$). From $\mathscr{G}$ we know that an onion tortilla is made of onion and, since it is a kind of tortilla, of potatoes and eggs. We know that "steak and chips" is a dish and a thing, and that it is made of beef and potatoes. We can extract common characteristics about "onion tortilla" and "steak and chips": both are dishes made of potatoes.

It should be noticed in this example that the predicate knowledge about a subsumer of g also applies to g. We can represent the predicate knowledge about a g by the set

$$\{(p, o) \mid \exists s : (s, p, o) \in E \text{ and } g \sqsubseteq s\}.$$

Each couple in this set corresponds to a knowledge unit about g, thought of as one of its characteristics. Some of these characteristics can be more specific than others. For instance, if we have "menu 1 is composed of strawberry ice-cream", then we also have that "menu 1 is composed of ice-cream", while the converse is not true. Thus we say that (`hcomp`, `strawberry i-c.`) is a more specific characteristic than (`hcomp`,`i-c.`). This intuition is formalized by the subsumption relation on $P \times V$ given by

$$(p_1, o_1) \sqsubseteq (p_2, o_2) \text{ if } p_1 \sqsubseteq p_2 \text{ and } o_1 \sqsubseteq o_2, \tag{2}$$

for $p_1, p_2 \in P$ and $o_1, o_2 \in V$. With this definition, $(p_1, o_1) \sqsubseteq (p_2, o_2)$ expresses the fact that (p_1, o_1) gives a more specific information than (p_2, o_2). Moreover, for $A \subseteq P \times V$, $mins(A)$ is the smallest set (w.r.t the subsumption relation on $P \times V$) that describes the same characteristics as A.

We are now able to provide a description function for elements of G, based on both subsumptional and predicate knowledge. The description function $\delta^1 : G \to D^{(1)}$, where $D^{(1)} = V \times 2^{P \times V}$, is defined by

$$\delta^1(g) =< g, mins(\{(p, o) \mid \exists s : (s, p, o) \in E \text{ and } g \sqsubseteq s\}) >,$$

where $mins$ is defined in terms of the order $\sqsubseteq$ over $P \times V$ given by (2). Note that $D^{(1)}$ is now used as the set of patterns. The similarity between two descriptions $d_1 =< s_1, e_1 >$ and $d_2 =< s_2, e_2 >$ belonging to $D^{(1)}$ is given by

$$d_1 \sqcap d_2 =< s_1 \sqcap s_2, mins(\{(p_1 \sqcap p_2, o_1 \sqcap o_2) \mid (p_1, o_2) \in e_1 \text{ and } (p_2, o_2) \in e_2\}) > .$$

The description $d_1 \sqcap d_2$ represents the characteristics that are common to both d_1 and d_2. We can see that the first component of the similarity is given by $s_1 \sqcap s_2$, namely the least common subsumer of s_1 and s_2 in $(V, \sqsubseteq)$. For the second component, we have all possible overlappings between couples from e_1 and e_2.

This similarity operator endows $D^{(1)}$ with a semilattice structure whose most general element is $< \top_V, \{\} >$. Therefore $\mathcal{P}^1 = (G, (D^{(1)}, \sqcap), \delta^1)$ is a pattern structure.

Example 3. To illustrate, let us compute the descriptions of menu 1 ($\mathtt{m}_1$) and menu 2 ($\mathtt{m}_2$).

$$\delta^1(\mathtt{m}_1) =< \mathtt{m}_1, \{(\mathtt{hcomp}, \mathtt{vanilla\ i\text{-}c.}), (\mathtt{hcomp}, \mathtt{s\&c})\} >,$$
$$\delta^1(\mathtt{m}_2) =< \mathtt{m}_2, \{(\mathtt{hcomp}, \mathtt{strawberry\ i\text{-}c.}), (\mathtt{hcomp}, \mathtt{salad})\} >,$$

The similarity between both menus is

$$\delta^1(\mathtt{m}_1) \sqcap \delta^1(\mathtt{m}_2) =< \top_V, \{(\mathtt{hcomp}, \mathtt{i\text{-}c.}), (\mathtt{hcomp}, \mathtt{dish})\} > .$$

From this similarity we obtain that menu 1 and menu 2 are two things ($\top_V$) that are composed of an ice-cream and a dish.

It could be argued that the pattern structure described in this subsection is not fully satisfying since the description function δ^1 may fail to capture some characteristics that appear "deeper" in the graph $\mathscr{G}$. This problem is already illustrated in Example 3, where we see that the fact that both menus contain a dish made of potatoes is not taken into account. This is due to the fact that the description of each menu expresses "composed of a dish that is steak and chips" or "composed of a dish that is onion tortilla", but not "composed of a dish that has potato as ingredient". To arrive at such level of specificity, in the next section we will examine deeper connections in the graph $\mathscr{G}$ through multi-level descriptions.

4.2 Multi-Level Descriptions

The principle of multi-level descriptions is to nest the descriptions of neighbors of g into the description of g. For instance, the multi-level description of a menu should contain the descriptions of the dishes that compose it. We define *k-level description functions* $\delta^k : G \to D^{(k)}$ where $D^{(k)} = V \times 2^{P \times D^{(k-1)}}$ and $k > 1$ is the maximal number of nestings that are allowed in the description, by:

$$\delta^k(g) =< g, mins(\{(p, \delta^{k-1}(o)) \mid \exists s : (s, p, o) \in E \text{ and } g \sqsubseteq s\}) > .$$

Note that $\delta^{k-1}(o)$ is the description of level $(k-1)$ of the neighbors of g in $\mathscr{G}$. The similarity between two descriptions $d_1 =< s_1, e_1 >$ and $d_2 =< s_2, e_2 >$ is given by

$$d_1 \sqcap d_2 =< s_1 \sqcap s_2, mins(\{(p_1 \sqcap p_2, d_1' \sqcap d_2') \mid (p_1, d_2') \in e_1 \text{ and } (p_2, d_2') \in e_2\}) > .$$

Again, this similarity operator endows $D^{(k)}$ with a semilattice structure whose most general element is $< \top_V, \{\} >$. Therefore $\mathcal{P}^k = (G, (D^{(k)}, \sqcap), \delta^k)$ is a pattern structure.

Example 4. Continuing from Example 3, the 2-level descriptions of menu 1 ($\mathtt{m}_1$) and menu 2 ($\mathtt{m}_2$) are

$$\delta^2(\mathtt{m}_1) =< \mathtt{m}_1, \{(\mathtt{hcomp}, < \mathtt{vanilla\ i\text{-}c.}, \{\} >),$$
$$(\mathtt{hcomp}, < \mathtt{s\&c}, \{(\mathtt{hing}, \mathtt{potato}), (\mathtt{hing}, \mathtt{beef})\} >)\} >,$$
$$\delta^2(\mathtt{m}_2) =< \mathtt{m}_1, \{(\mathtt{hcomp}, < \mathtt{strawberry\ i\text{-}c.}, \{\} >),$$
$$(\mathtt{hcomp}, < \mathtt{o.tortilla}, \{(\mathtt{hing}, \mathtt{potato}), (\mathtt{hing}, \mathtt{egg}), (\mathtt{hing}, \mathtt{onion})\} >)\} >,$$

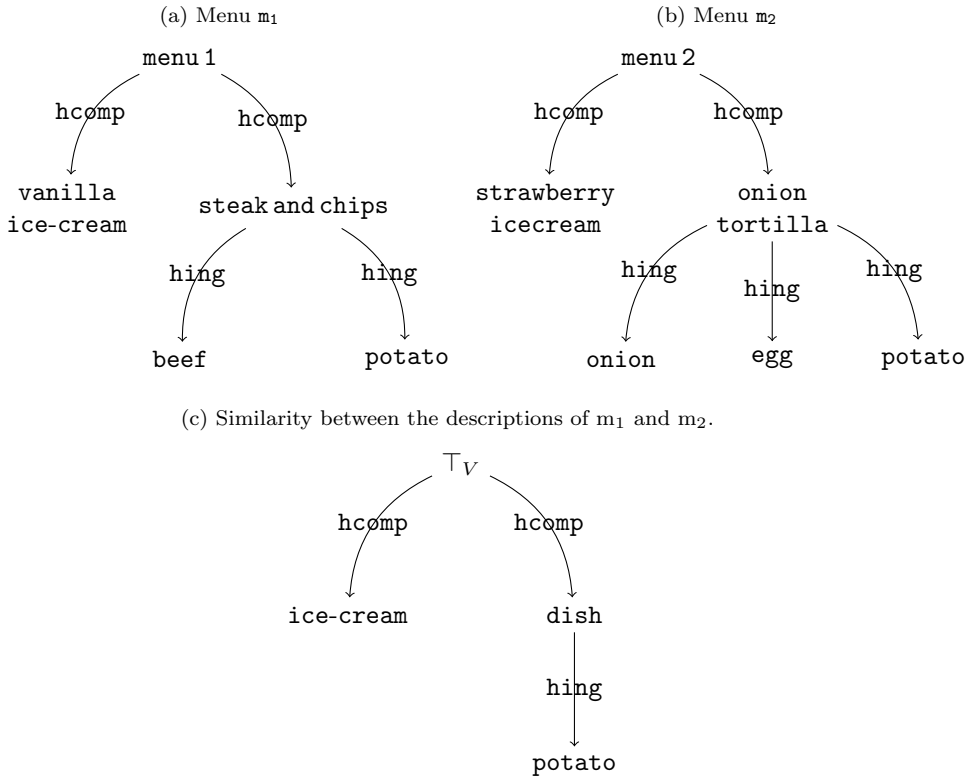

Figure 4: Trees representing the 2-level descriptions of $\mathtt{m}_1$ (4a) and $\mathtt{m}_2$ (4b), and the similarity of their descriptions (4c).

and the 2-level similarity between these descriptions is

$$\delta^2(\mathtt{m}_1) \sqcap \delta^2(\mathtt{m}_2) =< \top_V, \{(\mathtt{hcomp}, \mathtt{i\text{-}c.}), (\mathtt{hcomp}, < \mathtt{dish}, \{(\mathtt{hing}, \mathtt{potato})\} >)\} > .$$

The similarity between the 2-level descriptions of m_1 and m_2 expresses the fact that both are things composed of ice-cream and of a dish in which potato is an ingredient.

Interestingly, due to the recursive nature of k-level descriptions, they can be represented as trees of depth at most k. The tree corresponding to a description $d =< s, e >\in D^{(k)}$ can be drawn through the following steps:

1. The root of the tree has value s.
2. For all $(p, d') \in e$, create a branch labelled by p, leading to a child that is the subtree given by d'.

Example 5. The trees corresponding to $\delta^2(\mathtt{m}_1)$, $\delta^2(\mathtt{m}_2)$ and the similarity between these descriptions are given in Figure 4.

This pattern structure $\mathcal{P}^k$ is based on a similarity operation that the extraction of common characteristics of descriptions in a meaningful manner. Moreover the parameter k can be chosen by the user to control the deepness of

descriptions. As we have seen, deeper descriptions allow a more complete detection of common characteristics.

Even though the number of levels considered can be as high as desired, the size of descriptions could grow rapidly with the number of levels. This could constitute a drawback as such pattern structures could deemed to be unusable in practice. However, it is reasonable to assume that most relevant information about an object can be found in its vicinity, that is for small values of k.

5 Discussion and Perspectives

In this paper we proposed a method for building a pattern structure that exploits ontological knowledge in object descriptions, and that opens new perspectives in the classification of complex and structured data. Indeed, since pattern structures identify common characteristics of subsets of objects (through the similarity operation), they can be used to learn intentional definitions of target classes (see [2]). Since we are mainly interested in the classification of instances, the relational graph is only used as a source of information from which we build descriptions of the instances, that are represented as trees. This makes our approach different from those of [5, 11], that are based on FCA and pattern structures, but where the comparisons are made between graphs.

Also, in [8, 9] the authors propose a setting for ordinal classification, based on the theory of rough sets, and that makes use of ontological knowledge. This approach considers a generality/specificity relation between objects for aiding the preference prediction. However, this approach has two main limitations: it implicitly forces a correspondence between the preference and the generality relations, and it does not take into account all kinds of predicates available in the ontology. Our framework brings an alternative to this rough set approach that can even be used in situations where objects are relational structures such as trees, ordered structures or even graphs. For instance, when dealing with menus (seen as trees), the classification task could be that of classifying menus into preference classes, or with respect to suitable diets. Future work will also focus on application and empirical aspects, in order to evaluate the efficiency of this formalism in knowledge discovery and classification tasks.

References

[1] Alam, M., Napoli, A.: Interactive Exploration over RDF Data using Formal Concept Analysis. In: Proceedings of IEEE International Conference on Data Science and Advanced Analytics. Paris, France (Aug 2015)

[2] Belohlavek, R., De Baets, B., Outrata, J., Vychodil, V.: Inducing decision trees via concept lattices 1. International journal of general systems 38(4), 455–467 (2009)

[3] Caspard, N., Leclerc, B., Monjardet, B.: Finite ordered sets: concepts, results and uses. No. 144, Cambridge University Press (2012)

[4] Davey, B.A., Priestley, H.A.: Introduction to lattices and order. Cambridge university press (2002)

[5] Ganter, B., Grigoriev, P.A., Kuznetsov, S.O., Samokhin, M.V.: Concept-based data mining with scaled labeled graphs. In: International Conference on Conceptual Structures. vol. 3127, pp. 94–108. Springer (2004)

[6] Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: International Conference on Conceptual Structures. pp. 129–142. Springer (2001)

[7] Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edn. (1997)

[8] Pancerz, K.: Decision rules in simple decision systems over ontological graphs. In: Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013. pp. 111–120. Springer (2013)

[9] Pancerz, K., Lewicki, A., Tadeusiewicz, R.: Ant-based extraction of rules in simple decision systems over ontological graphs. International Journal of Applied Mathematics and Computer Science 25(2), 377–387 (2015)

[10] Reynaud, J., Toussaint, Y., Napoli, A.: Contribution to the Classification of Web of Data based on Formal Concept Analysis. In: What can FCA do for Artificial Intelligence (FCA4AI) (ECAI 2016). La Haye, Netherlands (Aug 2016)

[11] Soldano, H.: Extensional confluences and local closure operators. In: International Conference on Formal Concept Analysis. pp. 128–144. Springer (2015)